FIG. 1

102

T_1: {a, b, c, d, e, f, g}
T_2: {d, f, g}
T_3: {a, b, d, g}
T_4: {a, d, g}
T_5: {f, g}
T_6: {e, f, g}
T_7: {e, g}

108

EXAMPLE FOR DEPENDENCY (50% AS THRESHOLD)
{ag}
P(a | g) = COUNT(ag)/COUNT(g) = 3/7
P(g | a) = COUNT(ag)/COUNT(a) = 3/3
a->g, BUT NOT g->a
{ab}
P(a | b) = 2/2
P(b | a) = 2/3
a->b, AND b->a, (ab) IS NOT FREQUENT

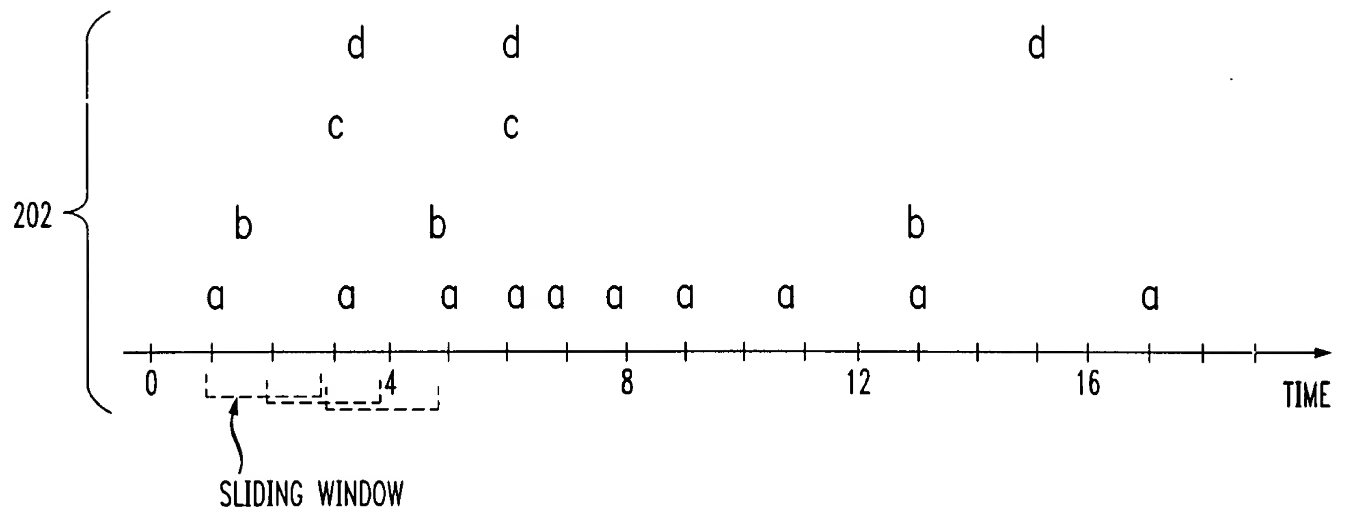
PATTERNS	COUNT
a	3
b	2
c	1
d	3
e	3
f	4
g	7

104

PATTERNS	COUNT
ab	2
ad	3
ae	1
af	1
ag	3
⋮	⋮

106

FIG. 2



PATTERNS	COUNT
a	10
b	3
c	2
d	3

204

PATTERNS	COUNT
ab	3
ac	2
dc	2
⋮	⋮

206

minsup = 3; minp = 0.6
{ab} IS FREQUENT, BUT NOT M-PATTERN
$P(a \mid b) = COUNT(ab)/COUNT(b) = 1;$
$P(b \mid a) = 3/10$
{dc} IS M-PATTERN, BUT NOT FREQUENT
$P(d \mid c) = 2/3;\ P(c \mid d) = 1$

208

FIG. 3

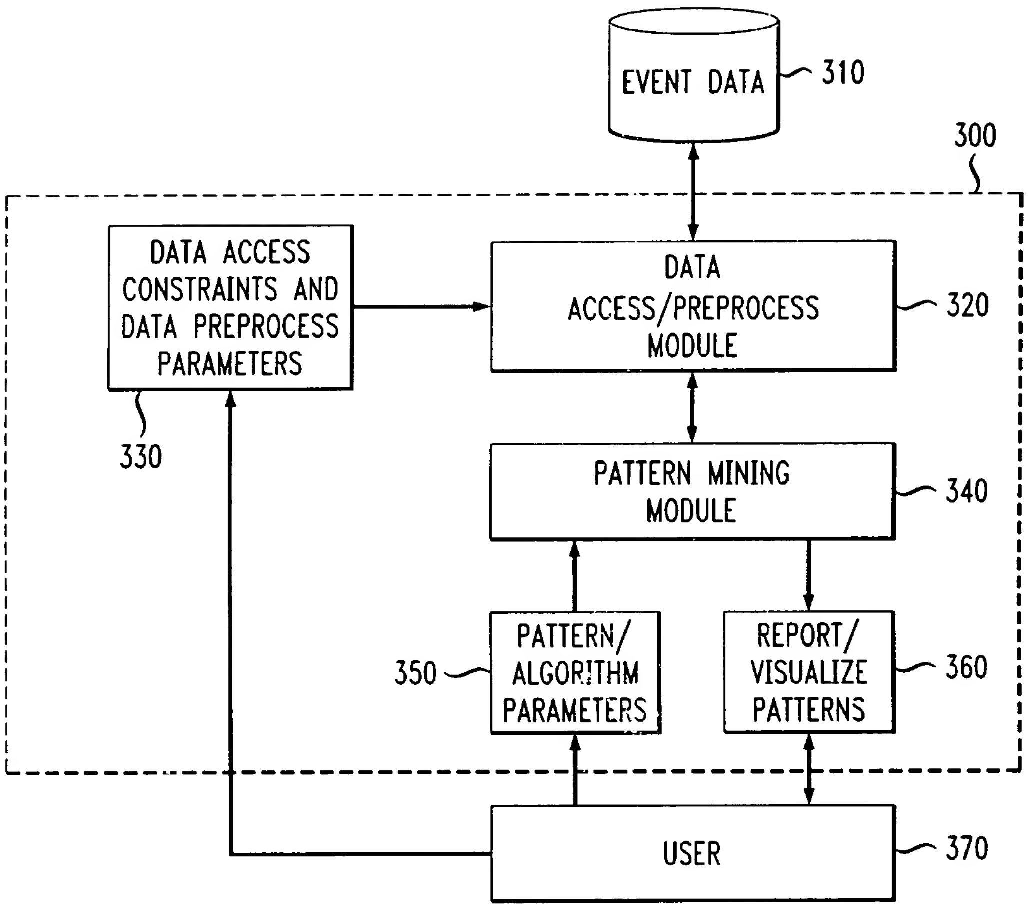
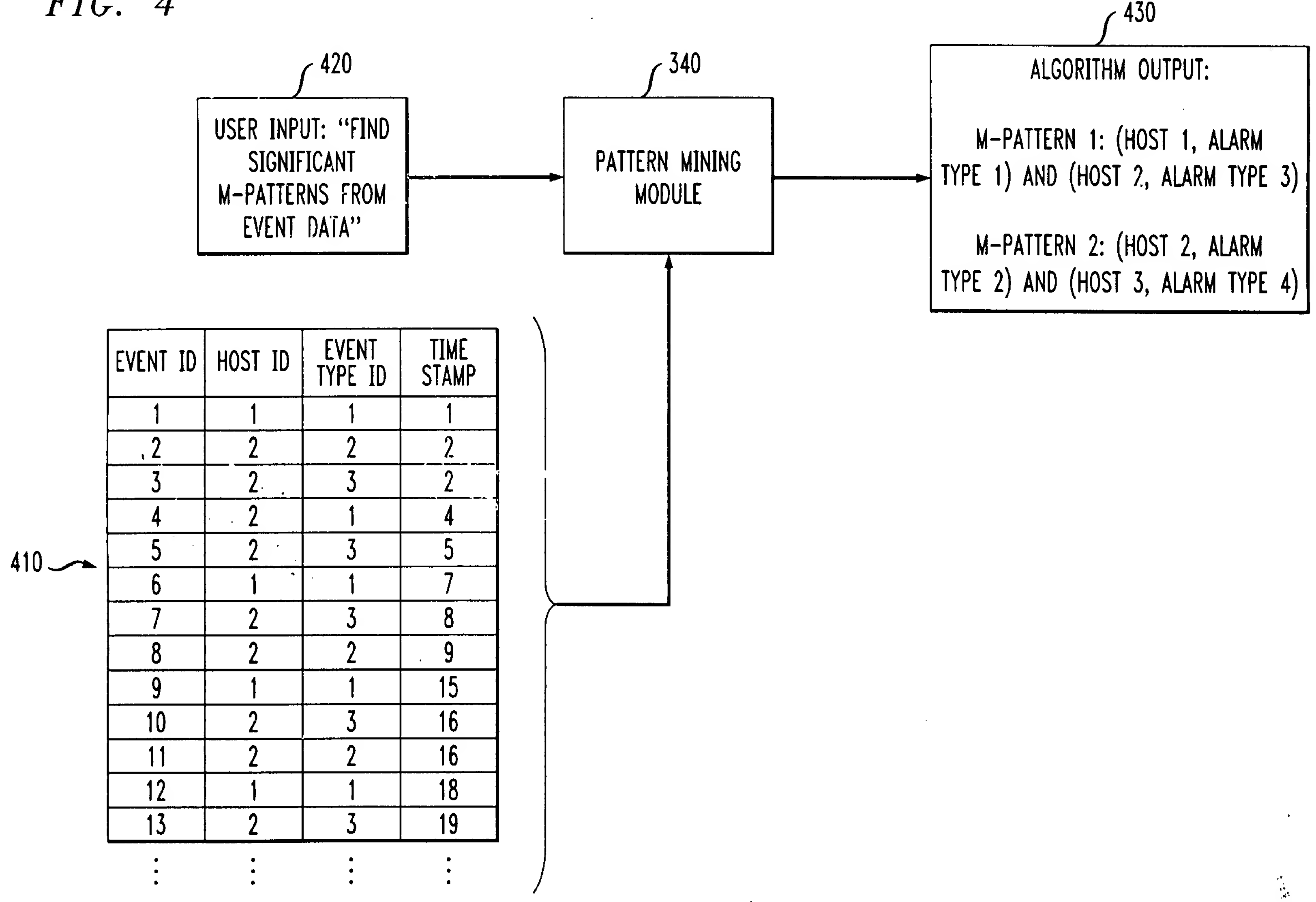

EVENT DATA
310
300
DATA ACCESS CONSTRAINTS AND DATA PREPROCESS PARAMETERS
330
DATA ACCESS/PREPROCESS MODULE
320
PATTERN MINING MODULE
340
PATTERN/ ALGORITHM PARAMETERS
350
REPORT/ VISUALIZE PATTERNS
360
USER
370

FIG. 4


420
USER INPUT: "FIND SIGNIFICANT M-PATTERNS FROM EVENT DATA"
340
PATTERN MINING MODULE
430
ALGORITHM OUTPUT:
M-PATTERN 1: (HOST 1, ALARM TYPE 1) AND (HOST 2, ALARM TYPE 3)
M-PATTERN 2: (HOST 2, ALARM TYPE 2) AND (HOST 3, ALARM TYPE 4)
410

EVENT ID	HOST ID	EVENT TYPE ID	TIME STAMP
1	1	1	1
2	2	2	2
3	2	3	2
4	2	1	4
5	2	3	5
6	1	1	7
7	2	3	8
8	2	2	9
9	1	1	15
10	2	3	16
11	2	2	16
12	1	1	18
13	2	3	19
⋮	⋮	⋮	⋮

FIG. 5

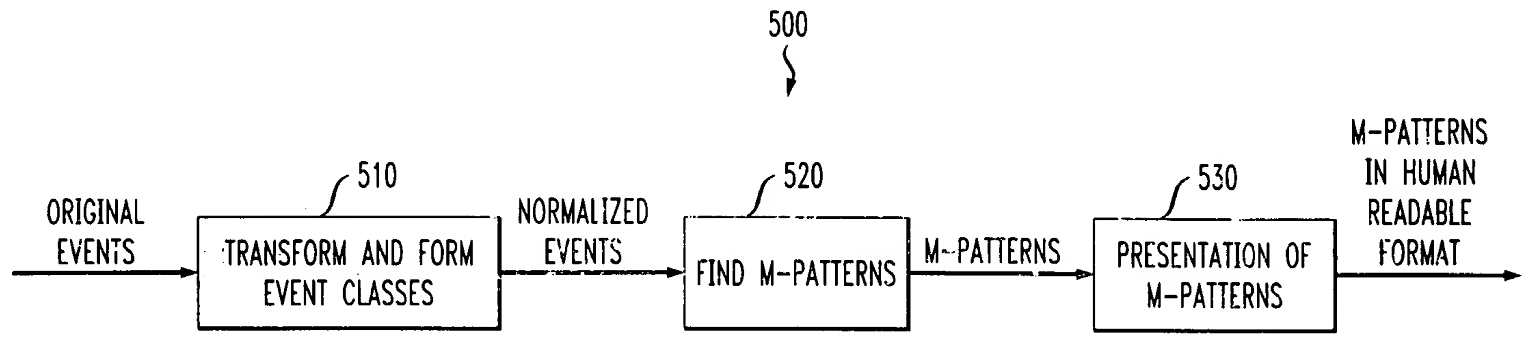
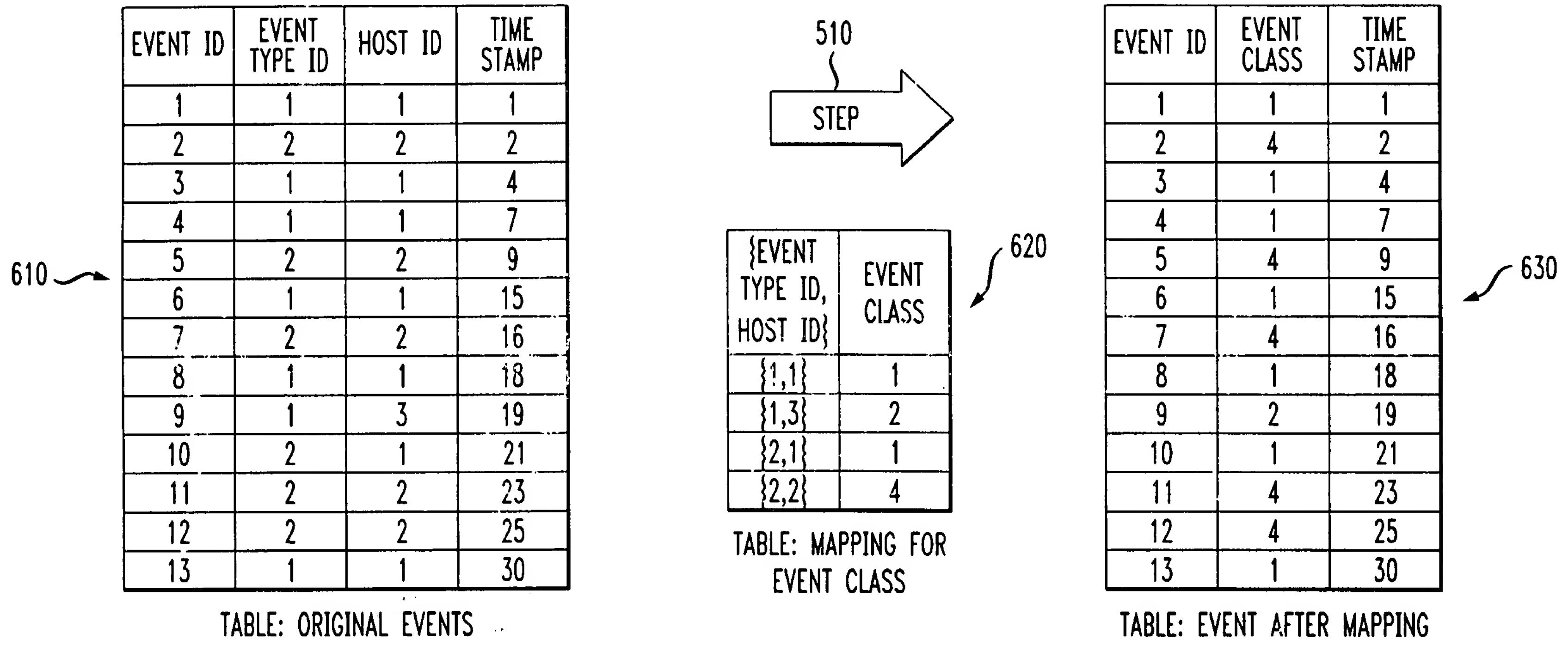

500
ORIGINAL EVENTS
510
TRANSFORM AND FORM EVENT CLASSES
NORMALIZED EVENTS
520
FIND M-PATTERNS
M-PATTERNS
530
PRESENTATION OF M-PATTERNS
M-PATTERNS IN HUMAN READABLE FORMAT

FIG. 6



EVENT ID	EVENT TYPE ID	HOST ID	TIME STAMP
1	1	1	1
2	2	2	2
3	1	1	4
4	1	1	7
5	2	2	9
6	1	1	15
7	2	2	16
8	1	1	18
9	1	3	19
10	2	1	21
11	2	2	23
12	2	2	25
13	1	1	30

TABLE: ORIGINAL EVENTS

{EVENT TYPE ID, HOST ID}	EVENT CLASS
{1,1}	1
{1,3}	2
{2,1}	1
{2,2}	4

TABLE: MAPPING FOR EVENT CLASS

EVENT ID	EVENT CLASS	TIME STAMP
1	1	1
2	4	2
3	1	4
4	1	7
5	4	9
6	1	15
7	4	16
8	1	18
9	2	19
10	1	21
11	4	23
12	4	25
13	1	30

TABLE: EVENT AFTER MAPPING

FIG. 7

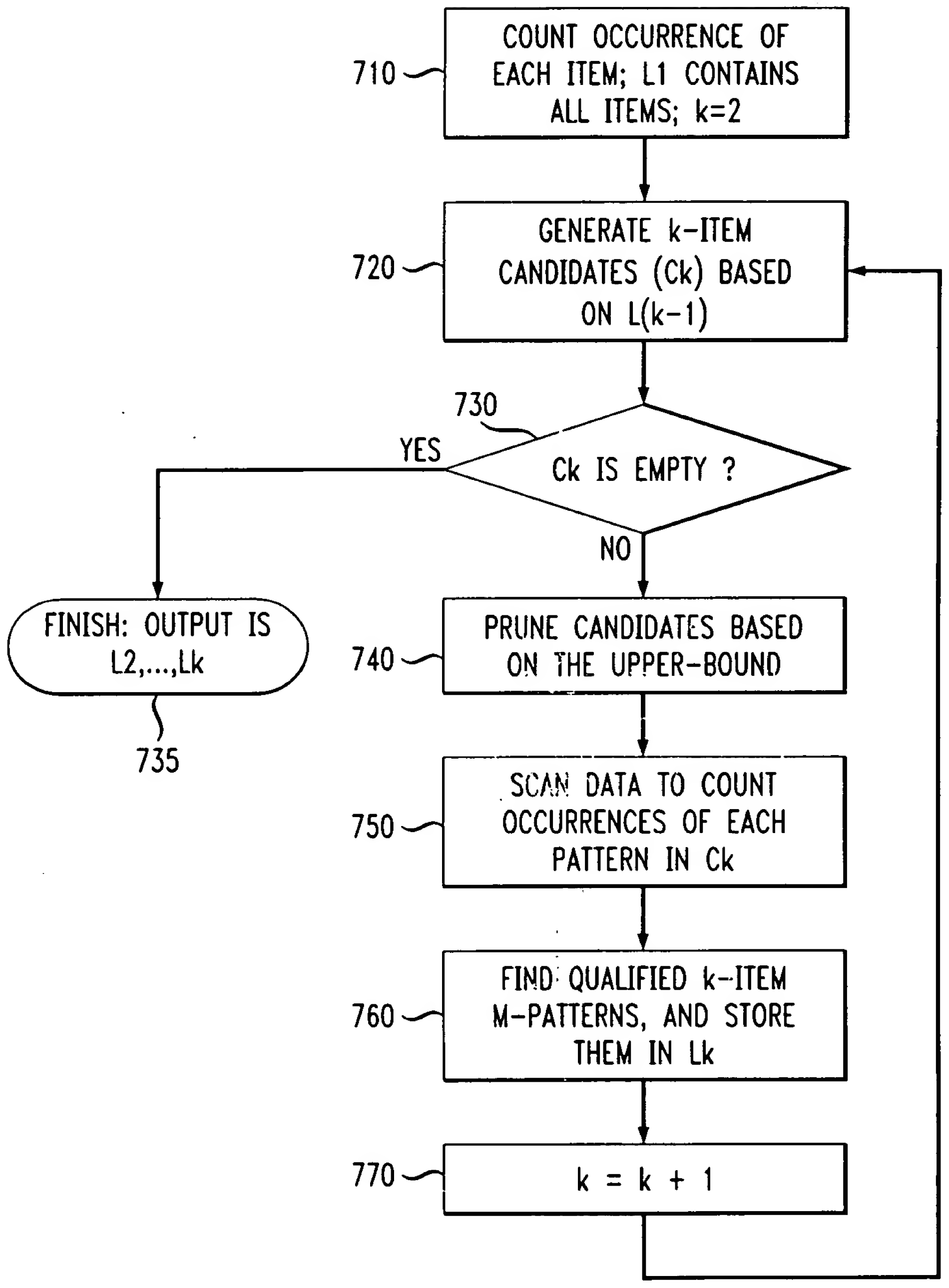

710
COUNT OCCURRENCE OF EACH ITEM; L1 CONTAINS ALL ITEMS; k=2
720
GENERATE k-ITEM CANDIDATES (Ck) BASED ON L(k-1)
730
Ck IS EMPTY ?
YES
NO
FINISH: OUTPUT IS L2,...,Lk
735
740
PRUNE CANDIDATES BASED ON THE UPPER-BOUND
750
SCAN DATA TO COUNT OCCURRENCES OF EACH PATTERN IN Ck
760
FIND QUALIFIED k-ITEM M-PATTERNS, AND STORE THEM IN Lk
770
k = k + 1

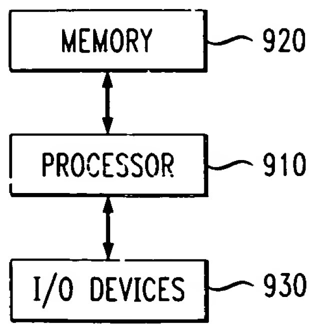
FIG. 8A

```
■ INPUT: A SET OF CANDIDATES $Ck$, COUNT INFORMATION AT ALL
  PREVIOUS LEVELS, AND A THRESHOLD minp
■ OUTPUT: A SET OF PRUNED CANDIDATES $C'k$
■ ALGORITHM
    • For each pattern $pat$ in $Ck$
        ▸ For each item $a$ in $pat$
            • Compute: $prob = Count(pat-a)/Count(a)$;
            ◦ if $prob < minp$
              - $Ck = Ck-pat$
              - break the for-loop
    • Return $Ck$
```

FIG. 8B

```
■ INPUT: PATTERN $pat$, ALL COUNT INFORMATION, AND A THRESHOLD $minp$
■ OUTPUT: TRUE IF $pat$ IS A QUALIFIED M-PATTERN; OTHERWISE FALSE.
■ ALGORITHM
    • For each $a$ in $pat$
        ▸ $prob = Count(pat)/Count(a)$
        ▸ if $prob < minp$
            • return false
    • Return true
■ This algorithm is O(k)
```

FIG. 9


MEMORY
920
PROCESSOR
910
I/O DEVICES
930